

Simultaneous Propulsion System and Trajectory Optimization

Eric S. Hendricks,* Robert D. Falck,[†] Justin S. Gray,[‡]

NASA Glenn Research Center, Cleveland, Ohio, 44135

A number of new aircraft concepts have recently been proposed which tightly couple the propulsion system design and operation with the overall vehicle design and performance characteristics. These concepts include propulsion technologies such as boundary layer ingestion, hybrid electric propulsion systems, distributed propulsion systems and variable cycle engines. Initial studies examining these concepts have typically used a traditional decoupled approach to aircraft design where the aerodynamics and propulsion designs are done a-priori and tabular data is used to provide inexpensive look up tables to the trajectory analysis. However the cost of generating the tabular data begins to grow exponentially when newer aircraft concepts require consideration of additional operational parameters such as multiple throttle settings, angle-of-attack effects on the propulsion system, or propulsion throttle setting effects on aerodynamics. This paper proposes a new modeling approach that eliminates the need to generate tabular data, instead allowing an expensive propulsion or aerodynamic analysis to be directly integrated into the trajectory analysis model enabling the entire design problem to be optimized in a fully coupled manner. The new method is demonstrated by implementing a canonical optimal control problem, the F-4 minimum time-to-climb trajectory optimization, using three relatively new analysis tools: OpenMDAO, PyCycle and Pointer. PyCycle and Pointer both provide analytic derivatives and OpenMDAO enables the two tools to be combined into a coupled model that can be run in an efficient parallel manner to offset the increased cost of the more expensive propulsion analysis. Results generated with this model serve as a validation of the tightly coupled design method and guide future studies to examine aircraft concepts with more complex operational dependencies for the aerodynamic and propulsion models.

Nomenclature

C	Aerodynamic cross force	β	Sideslip angle
D	Drag	γ	Flight path angle
F_n	Net thrust	λ	Azimuth angle
L	Lift	μ_{\oplus}	Gravitational parameter of Earth
M	Mach number	ω_{\oplus}	Rotation rate of Earth
r	Radius	ϕ	Latitude
T	Temperature	σ	Bank angle
$TSFC$	Thrust-specific fuel consumption	θ	Longitude
v	Airspeed		
α	Angle of attack		

I. Introduction

THE conceptual aircraft design process has traditionally used a mostly decoupled approach to determining the physical characteristics of the vehicle required to satisfy a set of mission performance requirements.^{1,2}

*Aerospace Engineer, Propulsion Systems Analysis Branch.

[†]Aerospace Engineer, Mission Analysis and Architecture Branch, AIAA Member

[‡]Aerospace Engineer, Propulsion Systems Analysis Branch, AIAA Member.

In this context, decoupled indicates that the major aircraft subsystems (fueselage, wings, propulsion, etc.) can be designed independently of each other and then integrated with their subsystem performance mostly unchanged. While this conceptual design process is sufficient for conventional aircraft concepts, many new propulsion concepts have been proposed which cannot be adequately assessed using a decoupled approach. These include boundary layer ingestion,^{3,4} hybrid electric propulsion systems,^{5,6} distributed propulsion systems⁷ and variable cycle engines.⁸ Boundary layer ingestion and distributed propulsion systems introduce strong propulsion-aerodynamic coupling that motivates a more coupled modeling approach.⁹ Hybrid electric propulsion systems, in addition to potential propulsion-aerodynamic coupling, can also introduce a strong coupling to the trajectory analysis as well. This coupling occurs as power split between the gas and electric power sources must be scheduled throughout a mission and the schedule has a strong impact on the overall design of the propulsion system.

While it is certainly possible to use a traditional, decoupled, aircraft design approach to get estimates of overall aircraft performance for designs including these new propulsion concepts, the accuracy of those estimates is limited by the lack of coupling in the models. In this work, a tightly coupled propulsion-trajectory approach that directly integrates a detailed 1-D thermodynamic propulsion model into the pseudospectral trajectory analysis is proposed to enable the design of more tightly integrated concepts. To validate this approach, a J79 turbojet engine with F-4 fighter aircraft is modeled and used in the solution of a canonical minimum time-to-climb trajectory optimization. The J79 model is implemented in PyCycle,¹⁰ a modular library of calculations that allows the user to build an arbitrary thermodynamic cycle analysis. The pseudospectral trajectory analysis is implemented in Pointer,¹¹ an implicit time integration library for numerically integrating ordinary differential equations. Both PyCycle and Pointer are written Python using NASA's OpenMDAO¹² framework and provide analytic derivatives to enable efficient gradient based optimization. One of the primary challenges with this new approach is that the detailed turbojet model is relatively computationally expensive—a single evaluation takes on the order of 20 seconds—compared to the traditional models built from regressions or tabular interpolations. The increased computation cost requires parallel computing in order to keep wall times reasonable for the optimization. To enable parallel execution, a Legendre-Gauss-Radau (LGR) transcription approach which is amenable to parallelization is employed. The combined J79 and F-4 model is then used to compare optimal trajectories for a series of engine designs with different after-burner temperature limits. The results show that the engine design has a strong impact on time-to-climb performance of the aircraft and can also have a meaningful affect on the shape of the final trajectory as well. Overall, the results prove the viability of this new approach which combines LGR pseudospectral trajectory analysis with the more expensive cycle analysis based propulsion model to create a coupled propulsion-trajectory analysis. The results also show that the method can be executed efficiently in parallel, thereby enabling future evaluation of more advanced, coupled aviation propulsion system concepts.

The remainder of the paper is organized as follows. First, brief overviews of OpenMDAO, PyCycle, and Pointer are presented. The selection of these analysis tools are crucial as they provide unique capabilities which facilitate development and application of the new method. Following the overviews of the selected analysis tools, the details of the new design method are presented. Next, an initial demonstration of the new method is then completed by applying it to the optimization of the F-4 minimum time-to-climb trajectory using several different configurations of the J79 engine. These results allow for an initial assessment of the method when applied to a conventional aircraft and engine design problem and a comparison of against the canonical solution to the F-4 minimum time-to-climb problem. Finally, notable conclusions are drawn and future work related to applying this design methodology to other operationally dependent technologies is discussed.

II. Analysis Tools

Development of the new design method proposed for operationally dependent concepts was facilitated by the use of two relatively new analysis tools, built inside the OpenMDAO framework: PyCycle and Pointer. These tools were selected because they each provide analytic derivatives enabling the use of gradient based optimization algorithms. Since both PyCycle and Pointer were each developed inside the OpenMDAO framework, it was the natural choice to also implement a coupled model in the same framework.

OpenMDAO

OpenMDAO provides the unifying framework for all the tools used in this work. It is an open source analysis framework developed at the NASA Glenn Research Center and written in the Python programming language.¹² It is designed to enable the multidisciplinary design, analysis and optimization of complex systems using gradient based optimization methods with analytic derivatives. In this capacity, it provides the ability to couple different disciplinary analysis tools, potentially written in different programming languages, into a single analysis model using a variety of different solvers and optimization methods.

One of the most important and unique features of OpenMDAO is its handling of derivative information within a multidisciplinary design, analysis and optimization problem. Description of these unique features is best completed using a notional MDAO problem as shown in Fig. 1. The objective in this example is to optimize F as a function of the design inputs x and state variables y . This requires executing three different disciplinary analyses as indicated by the blocks with C_1 and C_2 coupled by the state variables y . To use gradient based optimizer to solve this problem requires computing the total derivative $\frac{dF}{dx}$ which is made more difficult because of the coupling loop between C_1 and C_2 . This coupling loop prevents employing a simple chain rule to compute the total derivative, instead requiring computation of coupled derivatives by solving the unified derivative equations.¹³ OpenMDAO provides an efficient implementation of the unified derivative equations specifically tailored to the framework's modular approach to building coupled models. Each component is required to provide its own partial derivatives—derivatives of its outputs with respect to its own inputs—and then OpenMDAO will solve for the total derivatives across the whole model using either a direct or adjoint form for the best efficiency. The partial derivatives of each component can be computed either analytically or with finite difference approximation techniques and both methods can be mixed within a given multidisciplinary model.

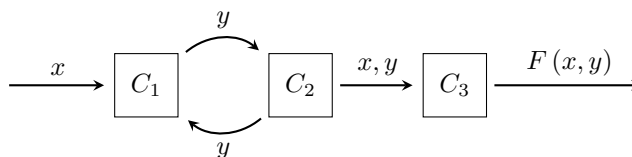


Figure 1: Notional MDAO Problem with Three Disciplinary Analyses.

Both PyCycle and Pointer are written natively in OpenMDAO by breaking the overall computations down into smaller sets of equations that are implemented as components. The components are then combined in specific arrangements to construct the overall computation. By keeping the set of equations in each component small, the task of deriving the partial derivatives is kept fairly simple. Then the framework handles the task of computing the total derivatives across the overall model without additional work from the user. Although OpenMDAO supports a wide range of optimizers, for this work all optimizations were performed using the gradient based SNOPT optimization package.¹⁴

PyCycle

The cycle analysis tool selected for this study is PyCycle which is a new analysis tool being developed at NASA Glenn Research Center. PyCycle models thermodynamics cycles using the same physics as the industry-standard NPSS tool, but its unique capability is that it provides analytic derivatives which is accomplished by taking advantage of the unified derivative equations implementation in OpenMDAO.

As described by Hearn et. al.,¹⁰ PyCycle breaks down the analysis of each engine element (inlet, compressor, burner, turbine, etc.) into a series of individual thermodynamic components and engineering components. Within these engine elements, the thermodynamic component calculations are performed using a chemical equilibrium method.¹⁵ The individual cycle elements can then be combined to form an overall cycle model as shown in Fig. 2. The engine shown in this figure is relatively simple turbojet with an afterburner representative of the J79 engine which will be used later in work. The blocks in the diagram represent the individual engine elements and the arrows indicate the major flow connections (f) along with the station's numeric designation. While the details of the calculations for most of these engine elements is outside the scope of this paper, there are several features which should be noted. First, the 'Ambient' element computes atmospheric properties using the U.S. 1976 standard atmosphere.¹⁶ Second, the 'Compressor' and 'Turbine'

elements employ interpolative maps to capture variations in turbomachinery efficiency and pressure ratio as a function of operating conditions such as shaft speed and mass flow. Furthermore, the ‘Compressor’ and ‘Turbine’ allow for cooling flows to be extracted and injected, respectively, at user specified locations within those elements. Lastly, the ‘Shaft’ element connects to the turbomachinery components to ensure the power required by the compressor is produced by the turbine. In addition to these basic engine elements, PyCycle models also include a Newton solver to converge the numerous chemical equilibrium calculations throughout each element as well as the higher level power balance on the shaft.

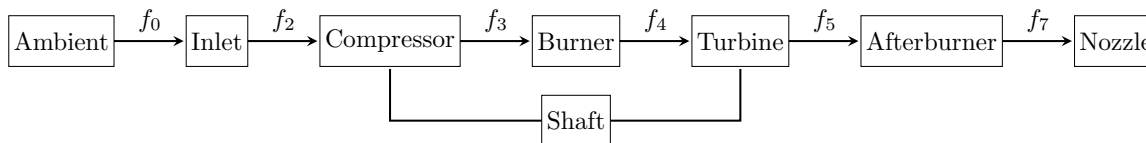


Figure 2: Flow Diagram of the J79 Turbojet Model in PyCycle.

Pointer

Pointer provides a general implementation of implicit time integration, using either Legendre-Gauss-Radau (LGR) pseudospectral^{17,18} or Legendre-Gauss-Lobatto (LGL)¹⁹ transcription, with a focus on solving general optimal-control problem formulations. Pointer’s optimization focused approach makes it well suited to implementation inside the OpenMDAO framework. The implicit time integration scheme discretizes the mission into a number of segments within which a polynomial is used to specify the value of each control or state parameter over time. The polynomials for each state equation are defined by a set of discretization nodes which are varied to satisfy the equations of motion (EOM) at collocation nodes on the function. By formulating the control problem in this way it can be solved with nonlinear programming approaches to determine the optimal control schedule.¹¹ The specifics of how the continuous-time optimal control problem is transcribed into a nonlinear programming problem is what distinguishes the LGR and LGL transcription methods. The general procedure for each approach is shown in Fig. 3. For this work, the LGR transcription was chosen because it uses a single stage evaluation process that parallelizes efficiently. For a given state polynomial order, both approaches use the same number of dynamics evaluations to compute the defects, but those of the LGR method can be fully parallelized, while the LGL approach uses an interpolation step between two separate evaluations of the dynamics. The two stage evaluation process of the dynamics in the LGL approach is an inherent parallel inefficiency. In the first stage, the EOM is evaluated at some number of discretization nodes which define the polynomial via Hermite interpolation. In the second stage, state values are interpolated to a second, smaller, set of collocation nodes. This means that, if the first stage is run fully in parallel, then up to two half of the processors are idle during the second stage.

III. Design Methodology

As described in the introduction, the traditional aircraft design process struggles with assessing new concepts which have tightly coupled propulsion-trajectory performance characteristics. The traditional approach to analyzing new concepts shown in Fig. 4 first designs the engine to a given set of thrust requirements (and possibly other secondary requirements as well). Next, off-design engine performance is tabulated across the flight envelope and possibly with an assumed schedule. This tabulated data is then passed into a separate trajectory optimization analysis that would interpolate it to compute engine performance as needed. Similarly the trajectory analysis will require airframe aerodynamic characteristics which may also be supplied in tabular form from an upstream aerodynamics analysis tool (e.g. a panel method or CFD solution). For both of these disciplines, generating the necessary tabular data is typically a matter of simply running sweeps over a limited number of parameters which define an operating envelope (e.g. Mach, altitude, and throttle setting for an engine). With a limited number of parameters, a full factorial exploration of the operating envelope is often computationally feasible with the number of samples equal to s^x where s is number of samples in each dimension and x is the number of dimensions.

However, for many of the advanced concepts being explored the propulsion and aerodynamic perfor-

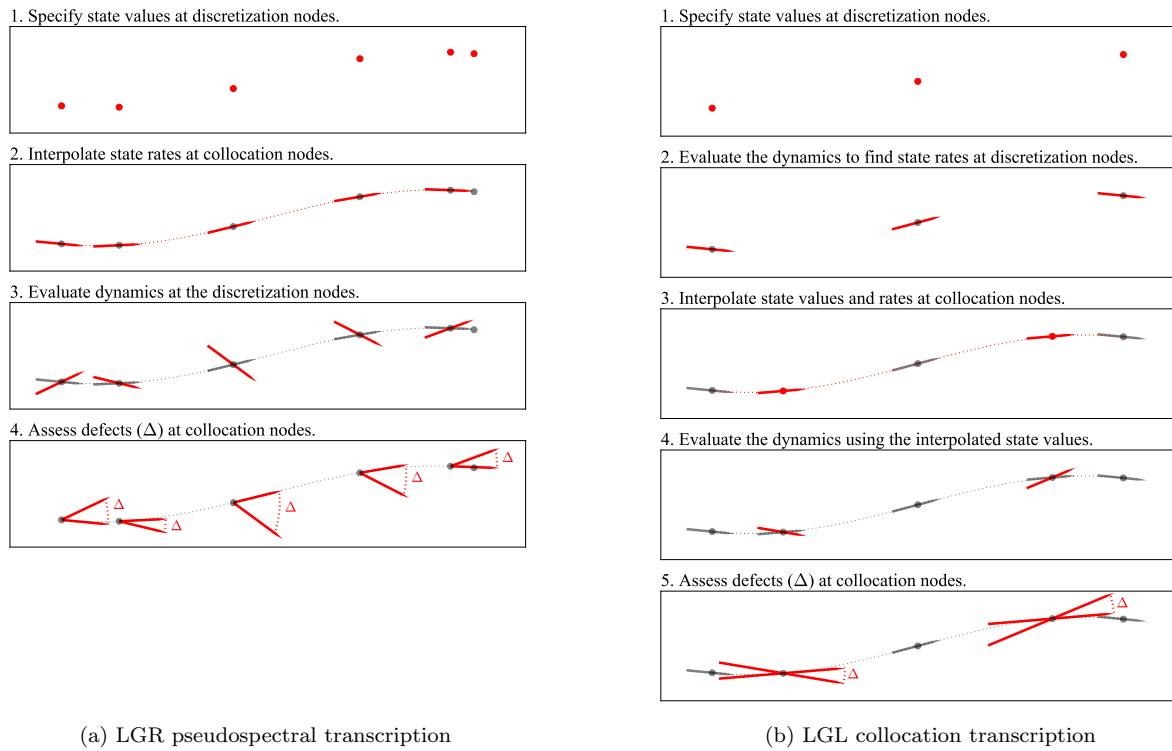


Figure 3: LGR Pseudospectral Transcription and LGL Collocation with 5th Order States.

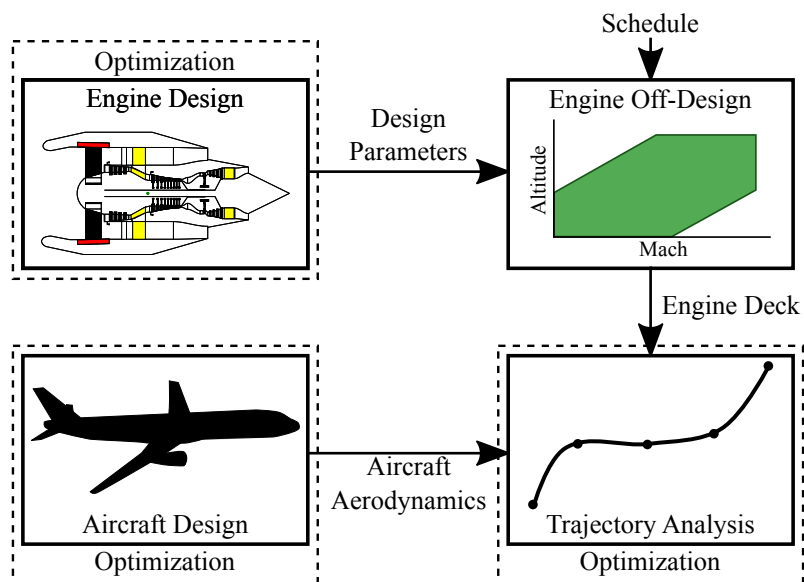


Figure 4: Traditional Engine and Trajectory Design Process.

mance characteristics are functions of larger sets of operating parameters. For example, a propulsion system with boundary layer ingestion might have performance that is also a function of angle-of-attack while the aircraft coupled to this engine might have drag dependent on the engine throttle setting. Similarly, a hybrid electric propulsion system will have two independent throttle settings—the overall throttle setting and the gas/electric split. And for some supersonic vehicles, variable cycle engines might have two or three different elements with schedules that affect overall performance (e.g. core driven fan stages or variable area nozzles). Geiselhart et.al.²⁰ noted the challenge of generating the tabular propulsion data in their work on the design of a supersonic aircraft, where they attempted to integrate an NPSS cycle analysis into the loop of their design optimization with just three independent dimensions. They identified problems with both the computational cost and the computational stability of NPSS which led to difficulties in generating the tabular data and heavily influenced their choice of optimization algorithms.

In these cases where the propulsion or aerodynamic characteristics are dependent on more than just a few operational parameters, generating the tabular data is often computationally infeasible as the dreaded ‘curse of dimensionality’ intervenes. In this so-called curse, the inclusion of additional operating parameters (x ’s) exponentially increases the number of cases required for a full factorial exploration of the operating envelope. While it may be possible to reduce the number of samples in each dimension or select an alternate sparse sampling strategy to reduce the number of points which must be evaluated, these approaches will likely suffer as interpolated values from these data sets may differ significantly from the actual performance.

To address this limitation, a new design methodology is proposed as shown in Fig. 5 to better support the analysis of unconventional concepts with larger numbers of independent operational variables. Overall, this method considers the same disciplinary analyses as the traditional process, but uses them in a more tightly coupled manner that avoids the need to fully enumerate the propulsion systems operating envelope as a pre-processing step for the trajectory analysis. The trajectory analysis, specifically using a pseudospectral method implemented in Pointer, only needs to know engine and aircraft performance at a relatively small set of discretization nodes as indicated in the figure. Rather than provide pre-generated tabular engine performance data that covers all possible operating conditions, the proposed approach allows the trajectory analysis (Pointer) to directly call the propulsion model (PyCycle) at each discretization node. In this formulation, there will consequently be the same number of a propulsion system off-design analysis points as the number of nodes in the trajectory analysis. A similar approach could be implemented for determining the aircraft aerodynamics with an expensive higher order model at a limited number of points then providing that data to the trajectory analysis. While the proposed integration approach is equally valid with a more advanced aerodynamic model, the focus of the current work is on the propulsion system and trajectory coupling. Therefore, the aerodynamic data considered in this work will still be provided by a simple drag polar formulation.

Directly coupling the engine and trajectory models requires passing information between the analyses at each node. Specifically, operating conditions, such as Mach number and altitude, and control settings, such as throttle setting, picked by the trajectory analysis are provided as inputs to the propulsion analyses. The propulsion analysis at each node then determines the engine performance characteristics such as thrust and fuel flow which are then passed back to the trajectory analysis. A separate instance of the propulsion model is created for each discretization node, which enables parallel execution of the individual nodes across multiple cores or multiple processors in a high performance computing environment.

IV. Methodology Demonstration on a Canonical Problem

To provide a demonstration of the proposed tightly coupled propulsion-trajectory optimization method, we implemented a canonical trajectory optimization problem: the F-4 minimum time-to-climb problem. This problem was first studied by Bryson and Denham²¹ in 1962 and has since been replicated in several other publications.^{22, 23} In summary, the problem aims to determine angle-of-attack and throttle settings that will get an F-4 aircraft from sea-level at Mach 0.34 to an altitude of 20,000 meters (approximately 65,000 feet) with a final speed of Mach 1.0.

The focus of the previous studies on the F-4 minimum time-to-climb problem was on determining the optimal trajectory (altitude, flight path angle, etc) that the aircraft should fly between the starting and ending conditions, and all assumed the propulsion system was operating at maximum throttle for the entire flight. The optimal trajectory was determined by varying only the angle of attack of the aircraft while the aircraft aerodynamics and propulsion system performance were modeled by regressions of manufacturer

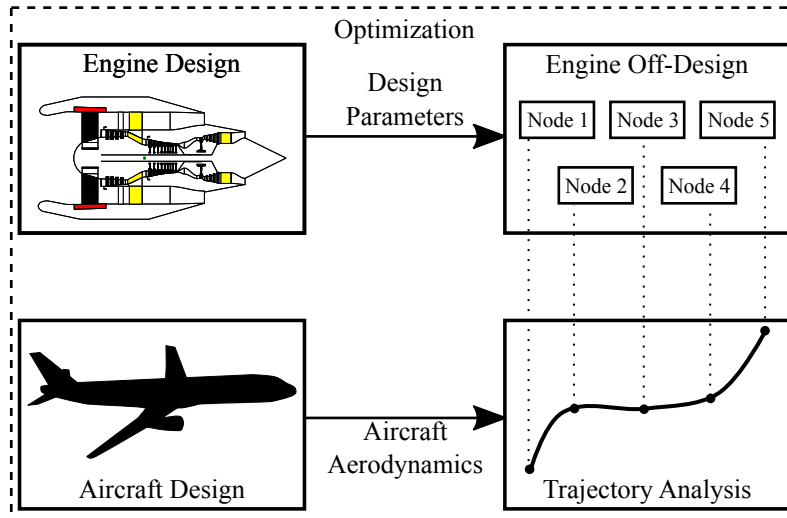


Figure 5: Proposed Engine and Trajectory Design Process.

data. Integration of more detailed and accurate models in these simulations was limited by the prohibitive computational cost and lack of parallel computing capabilities. The optimal trajectory solution found using these assumptions had a non-trivial flight profile and reached the target altitude in a time of 332 seconds.²³ One of the more interesting features of this trajectory is a short dive after the aircraft has reached an intermediate altitude. This dive is used to gain a gravity assist to accelerate the aircraft through transonic conditions due to a lack of engine thrust. This trajectory was later validated by an F-4 test flight in 1962 at Patuxent River Naval Air Station.²³

Interestingly, several months after this validation flight another F-4 Phantom (possibly a different variant of the aircraft with improved engines) set a significantly faster world record time to 20 kilometers of 178.5 seconds.²⁴ In an attempt to better match this world record performance, an updated propulsion model was created in PyCycle to capture the higher performance of later J79 engine models and used it in place of the original propulsion model. Creation and implementation of this model within the trajectory analysis also allows for variable engine performance as a result of two additional control variables which can be added to the optimal control problem: combustor exit temperature and afterburner Fuel-to-Air Ratio (FAR). With the incorporation of this engine model, a new problem formulation with additional control variables and constraints is necessary and is shown in Table 1.

In this formulation, the range, fuel weight, altitude, Mach number, combustor exit temperature, and afterburner FAR were varied at each of the discretization nodes to seek the fastest possible trajectory. It is notable that temperature is used as the control variable for the combustor but FAR is used for the afterburner. Temperature was selected (instead of FAR) for the combustor as it has a natural upper bound and lower bound and hence using it eliminates having to impose an additional non-linear constraint on combustor exit temperature. However, temperature cannot easily be used for the afterburner because the lower bound becomes a function of the turbine exit temperature. In other words, it would be possible for the optimizer to select an afterburner temperature that was too low and would require the cycle analysis to remove heat from the flow. Therefore, FAR is used for the afterburner with an additional non-linear constraint created for the afterburner exit temperature.

Before presenting the results produced by applying the new methodology to this problem, the next two sections will describe the propulsion system and aircraft models implemented in PyCycle and Pointer in more detail as well as provide some validation data to show that they are accurate representation of the J79 engine and F-4 aircraft respectively.

Table 1: Minimum Time-to-Climb Problem Formulation.

	Variable	Description
Minimize:	t_n	Time at last node
With respect to:	$r_c[n]$	Range at cardinal nodes
	$W_f[n]$	Fuel weight at cardinal nodes
	$h_c[n]$	Altitude at cardinal nodes
	$MN_c[n]$	Mach number at cardinal nodes
	$T_{4,c}[n]$	Combustor exit temperature at cardinal nodes
	$FAR_{ab,c}[n]$	Fuel-to-air ratio of afterburner at cardinal nodes
Subject to:	$T_{4,max}$	Maximum combustor exit temperature
	$T_{7,max}$	Maximum afterburner exit temperature
	$\Delta r_i[m]$	Range defects at interior nodes
	$\Delta T_{4,i}[m]$	Combustor exit temperature defects at interior nodes
	$\Delta FAR_{ab,i}[m]$	Afterburner FAR defects at interior nodes

Propulsion System Model

The F-4 Phantom II aircraft examined in the canonical minimum time-to-climb problem uses two General Electric J79 engines for propulsion, but as noted before different variants of the J79 were used at different times and resulted in variations in aircraft performance. In this study, the J79 model developed in PyCycle model was representative of the J79-10 which was used in the B version of the F-4 aircraft. The major engine components and connections considered in this model were shown previously in Fig. 2. In addition to these major flow and shaft connections, bleed flows were extracted from the exit of the compressor to cool the turbine. Further details and assumptions used in the engine model are listed in Table 2.

Table 2: J79 Design Characteristics and Assumptions.

Parameter	Value
SLS Airflow, lbm/s	170
SLS Pressure Ratio	13.5
Compressor Design Efficiency	0.83
Turbine Design Efficiency	0.86
Maximum Combustor Temperature, R	2370.0
Minimum Afterburner FAR	0.02
Maximum Afterburner Temperature, R	3400.0

Before coupling this engine model to the aircraft model in the trajectory optimization, the performance characteristics of the developed J79 PyCycle model were validated through comparison to published data.^{25,26} In particular, the net thrust and specific fuel consumption characteristics of the model at six flight conditions were examined as shown in Fig. 6. Across all flight conditions, the PyCycle J79 model generally matched the published data well in the non-afterburning portion of the operating regime (low thrust portions of the curves). The larger differences between model and published data at the higher elevations likely result from differences in the assumed compressor and turbine performance maps (relative to the actual component performance characteristics) as the engine moved away from the sea-level static point used in design.

When the model was evaluated with the afterburner on, however, the model predicted significantly lower fuel burn while the thrust values were reasonably close. This is indicated by the low TSFC values for the high thrust segments of the curves in Fig. 6. This error indicates additional refinements should be made to the PyCycle code itself and the parameters of this specific engine model to capture inefficiencies in the combustion process in the afterburner that would result in increased the fuel consumption. Despite the error in the fuel consumption, the PyCycle model was deemed to be sufficiently representative for use in optimization problem under consideration.

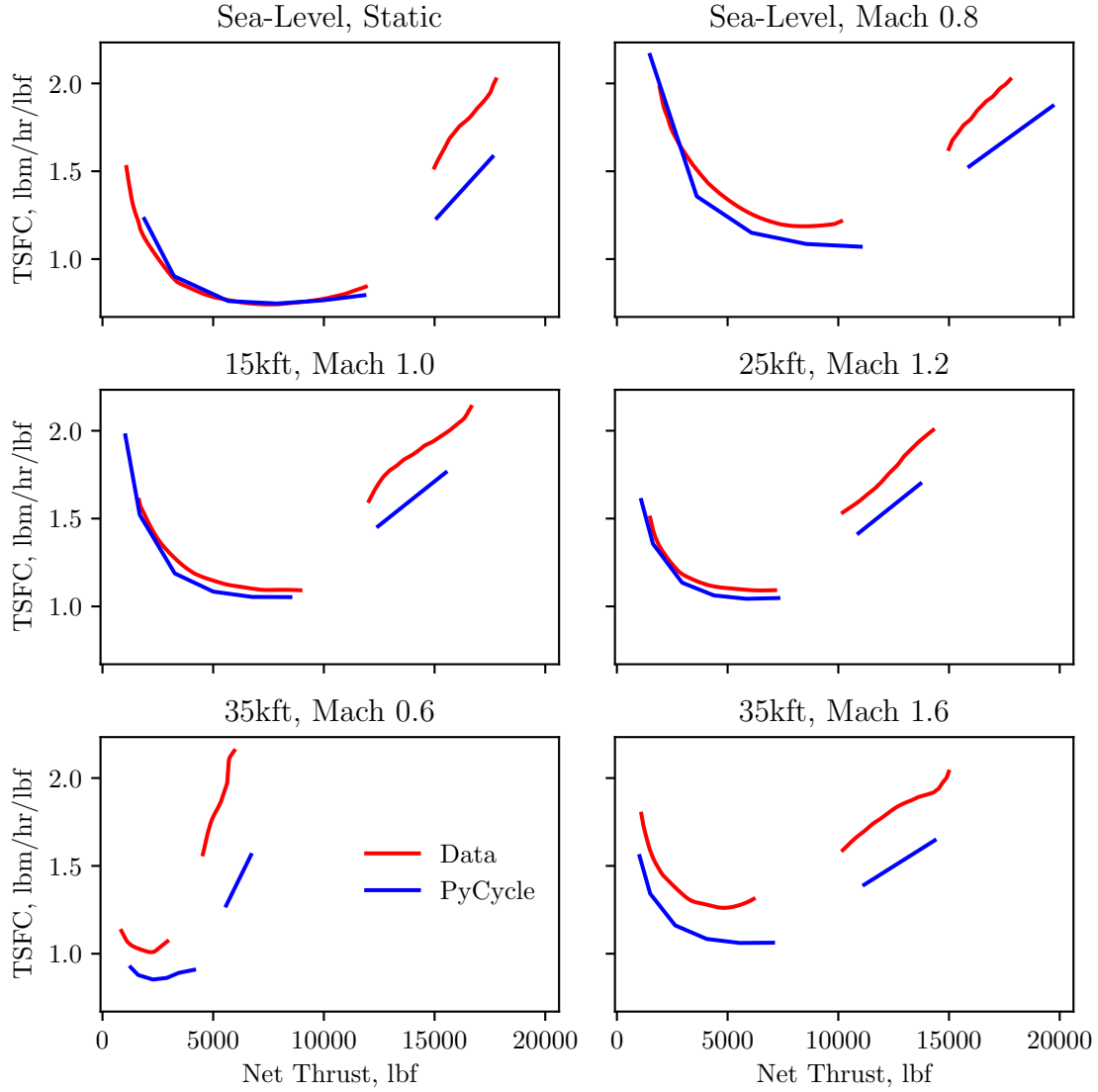


Figure 6: J79 Turbojet Model Performance.²⁶

Aircraft/Aerodynamics Model

For this study, the EOM for the F-4 fighter developed by Bryson²³ for the canonical minimum time-to-climb problem were used. The aircraft motion in the trajectory analysis is governed by the following three degree of freedom equations of motion for a body about a spherical Earth assuming no wind.

$$\begin{bmatrix} \dot{r} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \sin \gamma \\ \frac{\sin \lambda \cos \gamma}{r \cos \phi} \\ \frac{\cos \lambda \cos \gamma}{r} \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} \dot{v} \\ \dot{\lambda} \\ \dot{\gamma} \end{bmatrix} = \left(M_B^A \frac{(\bar{f}_{BA} + \bar{f}_{BP})}{m} + M_G^A (\bar{a}_G - \bar{v}_k) \right) \begin{bmatrix} 1 \\ \frac{1}{v \cos \gamma} \\ -\frac{1}{v} \end{bmatrix} \quad (2)$$

$$\dot{m} = -\dot{m}_{fuel} \quad (3)$$

The forces \bar{f}_{BA} and \bar{f}_{BP} represent the aerodynamic and propulsive forces in the body frame, respectively. Matrices M_G^A and M_B^A are used to transform from the local tangent plane and the vehicle body frame into the aerodynamic frame. In addition, it is assumed that the aircraft thrust vector is aligned with the body x-axis.

$$\bar{f}_{BA} = \begin{bmatrix} \sin \alpha & -\cos \alpha \cos \beta & \cos \alpha \sin \beta \\ 0 & -\sin \beta & -\cos \beta \\ -\cos \alpha & -\sin \alpha \cos \beta & -\sin \alpha \sin \beta \end{bmatrix} \begin{bmatrix} L \\ D \\ C \end{bmatrix} \quad (4)$$

$$\bar{f}_{BP} = \begin{bmatrix} F_n \\ 0 \\ 0 \end{bmatrix} \quad (5)$$

The term \bar{a}_G is the acceleration due to gravity in the local tangent plane. Here, a simple point-mass model for gravity is assumed.

$$\bar{a}_G = \begin{bmatrix} 0 \\ 0 \\ -\frac{\mu_\oplus}{r^2} \end{bmatrix} \quad (6)$$

The term \bar{v}_k is the acceleration term due to the rotation of the Earth:

$$\bar{v}_k = 2\omega_\oplus v \begin{bmatrix} \cos \gamma \sin \lambda \sin \phi \\ \cos \phi \sin \gamma - \cos \gamma \cos \lambda \sin \phi \\ \cos \phi \cos \gamma \sin \lambda \end{bmatrix} + r\omega_\oplus^2 \cos \phi \begin{bmatrix} \sin \phi \\ 0 \\ \cos \phi \end{bmatrix} + v^2 \frac{\cos \gamma}{r} \begin{bmatrix} \sin \gamma \cos \lambda + \sin^2 \lambda \cos \gamma \tan \phi \\ \sin \lambda (\sin \gamma - \cos \lambda \cos \gamma \tan \phi) \\ \cos \gamma \end{bmatrix} \quad (7)$$

For this problem it is also assumed the sideslip angle (β) and bank angle (σ) to be zero throughout the flight. The lift, drag, and cross force coefficients used to compute the forces in (4) are:

$$C_L = \alpha C_{L\alpha}(M) \quad (8)$$

$$C_D = C_{D0}(M) + \alpha C_{L\eta}(M) \quad (9)$$

$$C_C = 0 \quad (10)$$

Lastly, atmospheric properties are computed from the 1976 standard atmosphere.

As stated earlier, these equations of motion were used by Bryson and Denham in the first F-4 minimum time-to-climb solution along with a simple polynomial regression model for the propulsion system. To evaluate the Pointer implementation, these same equations of motion and the simple propulsion model in Pointer were used to reproduce Bryson's canonical results. The Pointer generated results replicate those produced by Bryson, achieving a similar time-to-climb of 315 seconds by following a similar trajectory as shown in Fig. 7. While the two implementations generally produce the same results, there is a minor discrepancy in the trajectory during the initial climb-out. This discrepancy is due to differences in the aerodynamic and atmospheric models between the two implementations. Despite these differences, the two trajectories share key characteristics such as a transonic dive and final zoom climb which validates the implementation of the F-4 equations of motion in Pointer.

Results of Trajectory Optimization with Baseline Engine

The maximum afterburner temperature of 3400 degrees Rankine, from Table 2, is of specific note because this value gave the best match to the published data that the engine model was validated against. This engine configuration was considered the baseline engine for this study. The trajectory optimization produced approximately a 200 second time-to-climb for the baseline engine as shown in Fig. 8. The aircraft accelerates

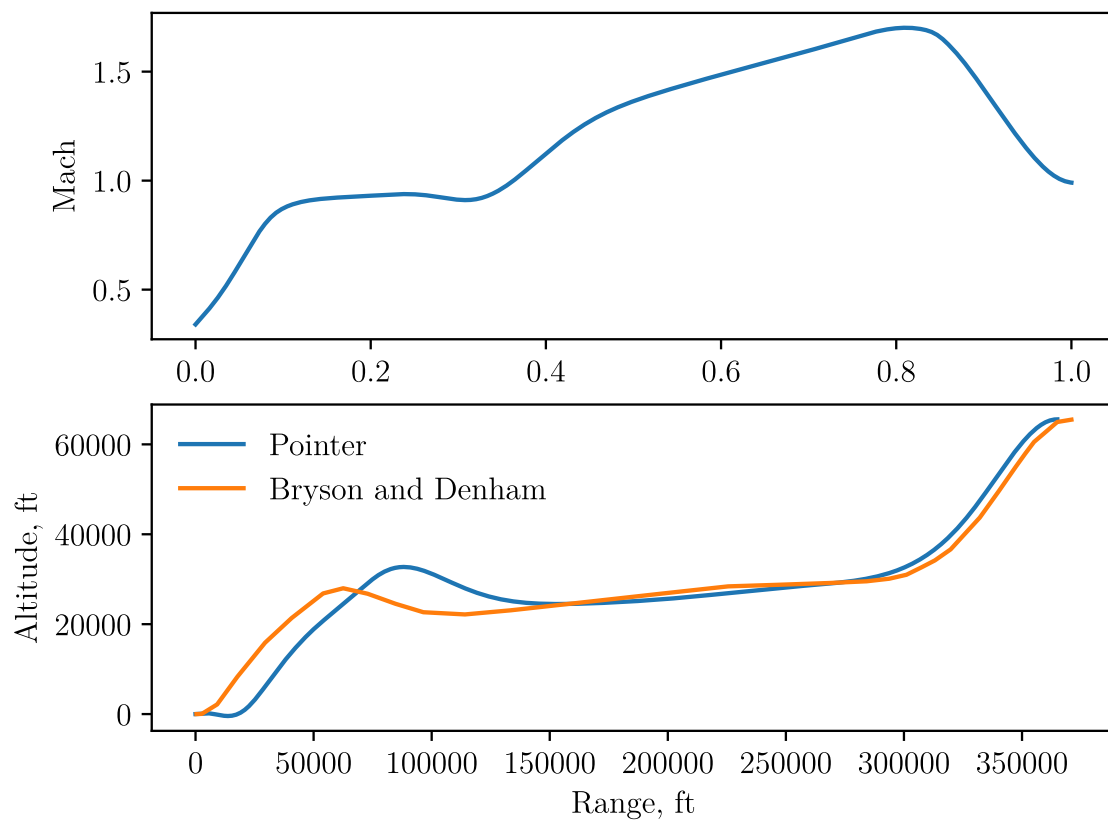


Figure 7: Optimal Trajectories for the F-4 Minimum Time-to-Climb Problem.²³

to just below sonic conditions near sea-level for the first 20 to 25 seconds, then holds approximately a constant Mach number as it climbs to an altitude of about 20,000 feet. At this point the trajectory starts to differ significantly from the one in Fig. 7. The previous solutions both showed a dive through transonic, but the new solution has a level acceleration phase until the aircraft reaches approximately Mach 1.7. Finally, the aircraft does a zoom climb to reach the target altitude of 65,000 feet in level, Mach 1 flight. The final time-to-climb is 203.6 seconds.

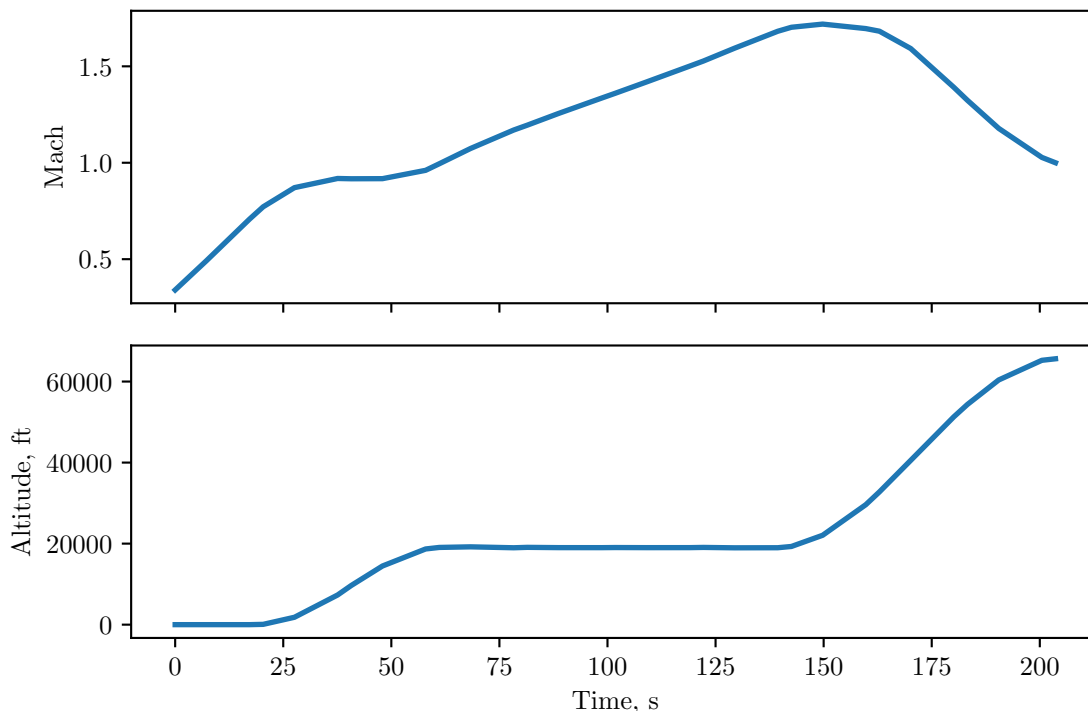


Figure 8: F-4 Minimum Time-to-Climb Trajectory Results.

In addition to evaluating the optimal trajectory, the identified optimal engine performance characteristics were examined. The optimizer determined that the engine should be run at the maximum combustor and afterburner temperatures throughout the flight. This result is expected as maximizing the temperature in both burners produces the most thrust for the engine leading to the best climb performance, and it is encouraging that the optimizer finds the expected result. Despite running at nearly constant combustor and afterburner temperatures however, the thrust produced by the engines varied throughout the flight as shown in Fig. 9. This variation is predominantly due to the changes in freestream Mach number and altitude over the course of the flight.

The lack of a dive through transonic and 100 second faster time-to-climb both indicate that the PyCycle J79 model is delivering significantly more thrust than the original simplified model. The final time of 203.6 seconds is also significantly closer to the official world record time-to-climb of 178.5 seconds. Since the aircraft equations of motion are identical between the validation case and the case with the baseline PyCycle model, the differences in performance must be from differences in the engine model. Since the result of this trajectory optimization still under performed the official record by 25 seconds, additional simulations were done with variations on the engine design to try and get closer agreement.

Results of Trajectory Optimization with Modified Engines

In an attempt to better match the world record time and demonstrate the capabilities of the method, the trajectory optimization was executed with several modified engine models. These modified models kept the core of the engine (i.e. the compressor, combustor and turbine) constant while changing only the operating

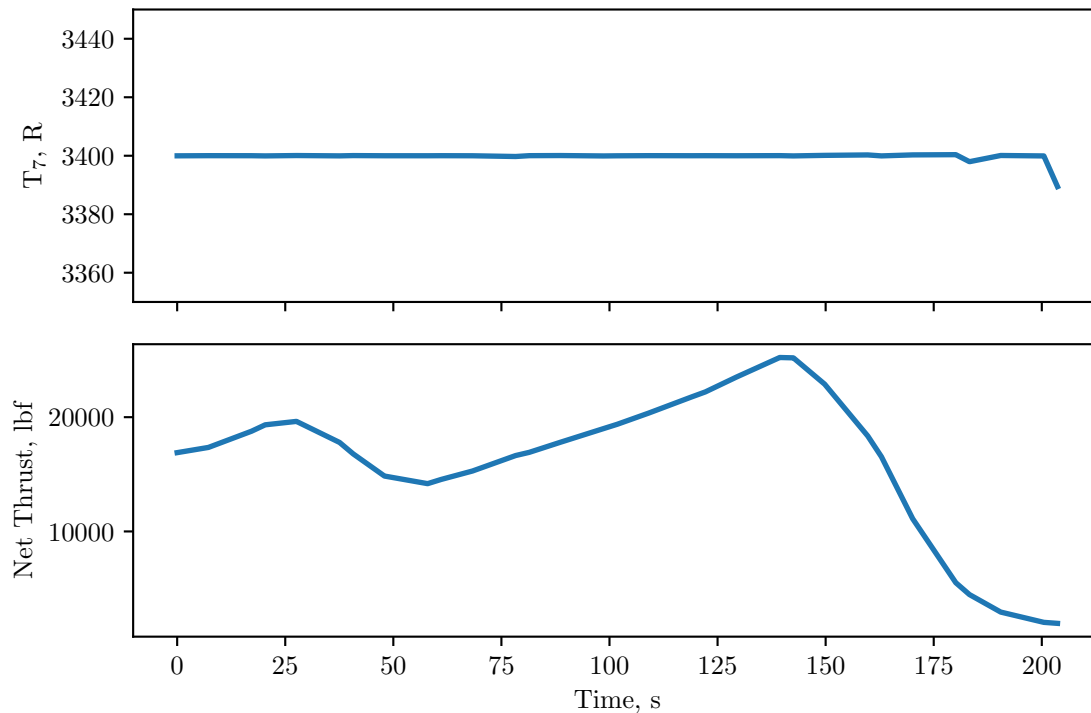


Figure 9: F-4 Minimum Time-to-Climb Engine Performance Results.

assumptions and limits of the afterburner. Modification of the afterburner was deemed to be the most appropriate alteration as the engine calibration to published data was weakest in this regime.

Two additional analyses were completed with different operating assumptions and limits on the afterburner. In the first case, the allowable afterburner exit temperature was increased to 3650 Rankine. For the second case, the afterburner limit was changed from temperature to a maximum fuel-to-air (FAR) ratio. A FAR limit of 0.04 was selected as that value was slightly below stoichiometric. Each of these cases was again run with the same number of discretization nodes, initial guess and optimization algorithm used to generate the baseline and validation results.

The resulting trajectories for these additional cases are shown in Fig. 10 along with the baseline solution. Both of the new trajectories follow similar flight paths as the initial solution with the aircraft completing an initial level acceleration then climbing to an intermediate altitude for a second level acceleration before climbing to the target flight condition. The major difference between the various results is that the aircraft reaches the target altitude and Mach number in less time than the baseline solution. The faster time-to-climb is a result of the increased thrust in these modified engines as shown in Fig. 11. Figure 11 also shows the afterburner temperatures determined by the optimizer throughout the flight.

Overall, the optimization with the two modified engines produced faster times to altitude with similar trajectories. The first case with an afterburner temperature limit of 3650 Rankine reached the target flight condition in 180.2 seconds while the FAR limited case came in at 164.2 seconds. These results indicate that an afterburner temperature limit of about 3650 Rankine may be more in line with the actual engine performance for the J79 engine used in the F-4 Phantom. Additionally, these results demonstrated the flexibility of the developed design methodology. The method was quickly able to adapt to revised engine performance models and selected the optimal control schedule to minimize the time-to-climb while staying within the specified operating limits.

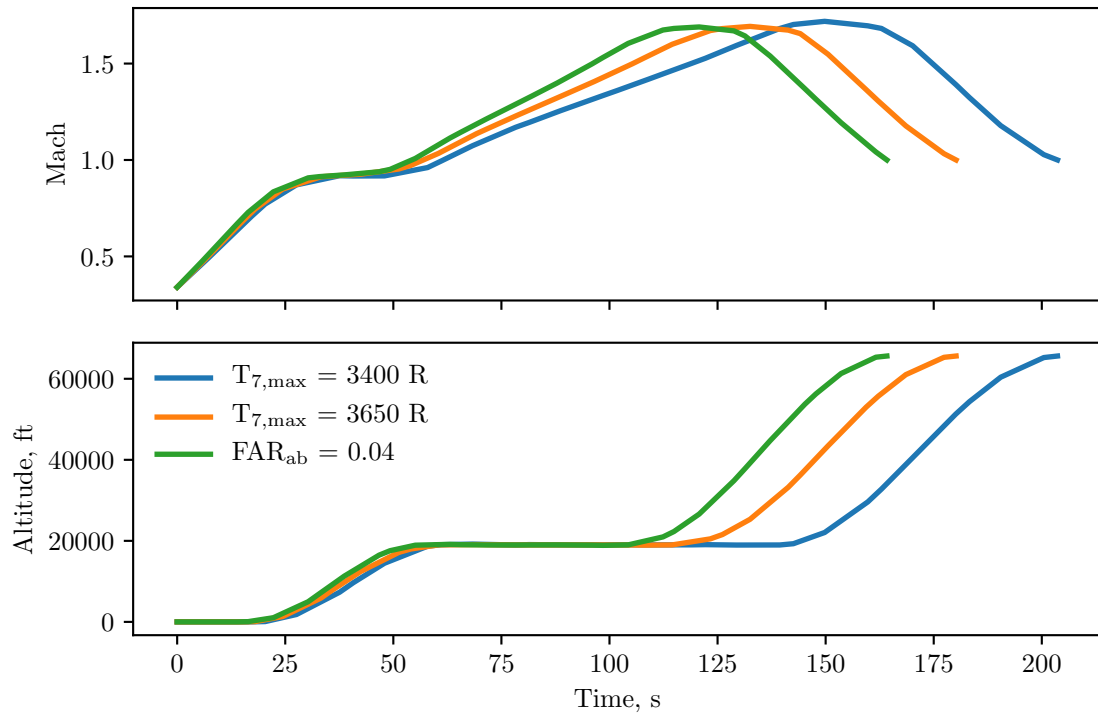


Figure 10: F-4 Minimum Time-to-Climb Trajectory Results with Modified Engines.

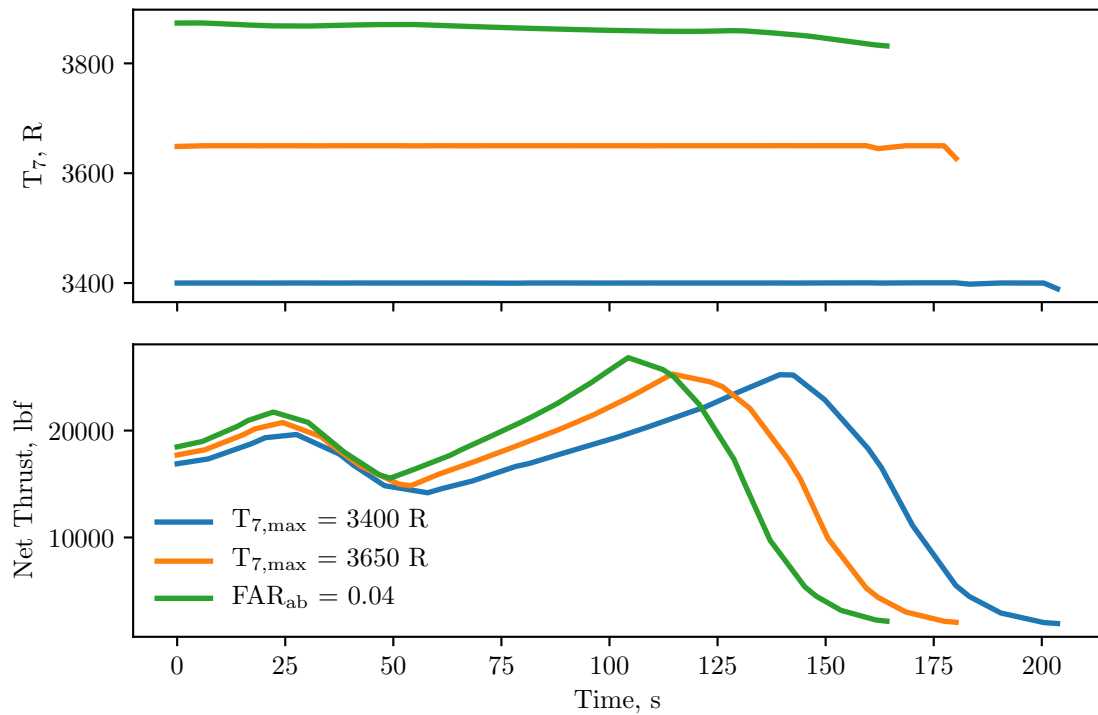


Figure 11: F-4 Minimum Time-to-Climb Engine Performance Results with Modified Engines.

Parallel Execution Performance Results

Due to the increased computational cost incurred by tightly coupling the cycle model directly, the ability to execute all the calls to the propulsion model in parallel is very important in order to keep the wall times reasonable. The LGR transcription approach was specifically selected for the high degree of parallelism inherent in its formulation. For the results presented in the previous sections, the trajectory was split up into 10 segments with 4 discretization nodes each, giving a total of 40 nodes. The distribution of these nodes is shown in Fig. 12, with each segment shown in an alternating color. Note that the endpoints of each segment overlap each other for the interior segments.

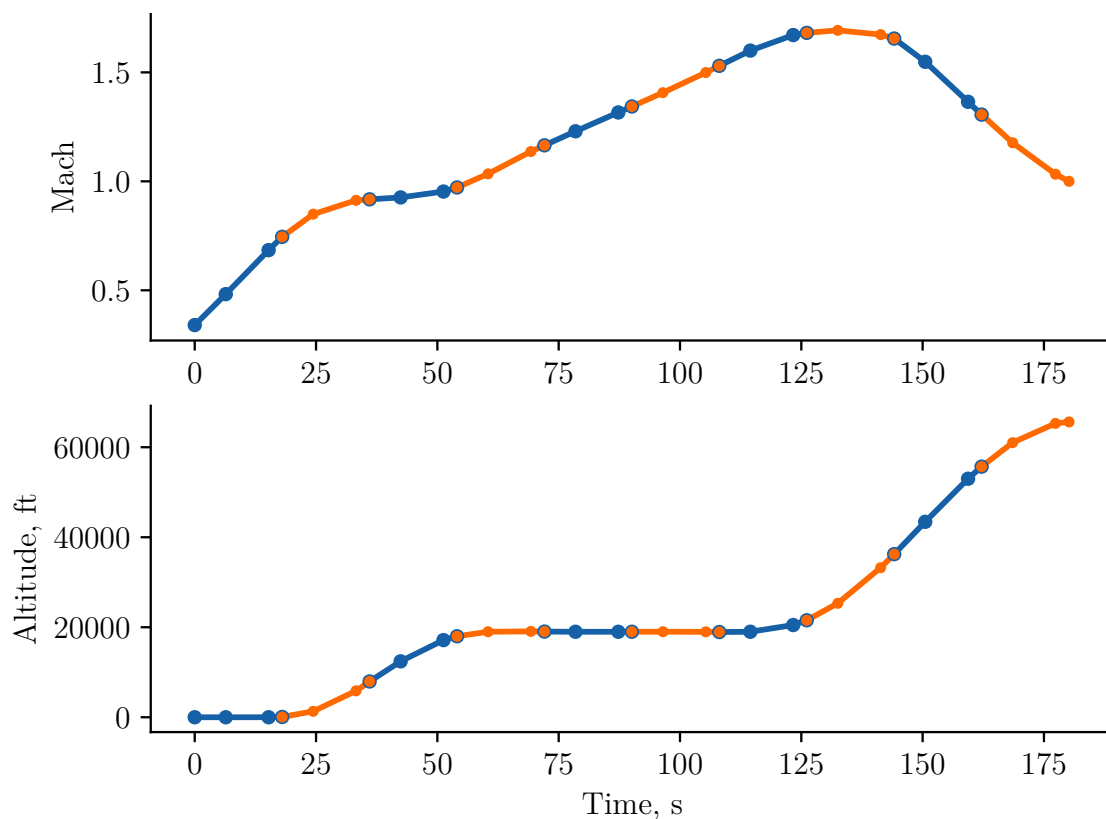


Figure 12: Collocation Segments and Nodes for the Optimal Trajectory.

Since there are 40 discretization nodes the model can be run most efficiently in parallel with any whole multiple of up to 40 processors. The model was timed with 1, 5, 10, 20, and 40 processors to assess its parallel performance. Parallelization was implemented using the built MPI based features of the OpenMDAO framework, and did not require any additional MPI code to be added to the analyses themselves. The parallel efficiency was tested by recording the time to run a single pass through the evaluation model on each of these processor amounts, and the results are shown in Fig. 13. The model scales nearly ideally up to 10 processors, but the performance begins to fall off somewhat at 20 and 40 processors. At these higher processor counts, slight serial bottle necks in the calculations have more impact on the parallel efficiency and become more apparent. Furthermore, the cost of converging the propulsion cycle models for each point is not exactly equal causing the calculation to be held up by the slowest converging point.

V. Conclusion and Future Work

Numerous advanced aircraft concepts have been proposed for addressing future challenges in aviation. Many of these advanced concepts use novel propulsion system designs such as boundary layer ingestion,

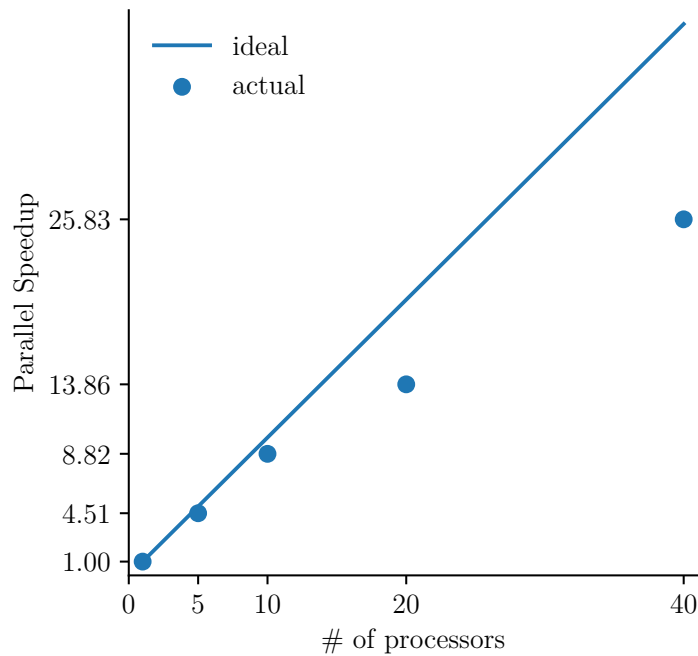


Figure 13: Parallel Scaling of the Combined PyCycle and Pointer Model.

variable cycle engines, or hybrid electric cycles that depended on innovative integration within the aircraft. The design and sizing of these systems is tightly coupled to the trajectory flown by the aircraft and the associated control schedule required throughout the flight.

To support the emerging design and analysis needs for these concepts, a new modeling method was developed which tightly couples a full propulsion model with a trajectory analyses. This is achieved by the trajectory analysis directly calling the propulsion model instead of the more traditional process of generating tabular engine data as a pre-processing step that then gets interpolated by the trajectory analysis. Since the propulsion model is significantly more computationally expensive than a simple interpolation, parallel computing was required to keep the wall times reasonable for the optimizations and a pseudospectral time integration scheme with an LGR based transcription was selected specifically for its parallelizable qualities.

To evaluate this new method, it was applied to a canonical minimum time-to-climb problem for an F-4 fighter aircraft. Three different engine designs were considered, with varying limiting conditions in the afterburner that changed the maximum thrust the engine could produce. The trajectory and engine control optimization completed in these assessments generally produced similar trajectories for all three engines considered. The primary difference in the resulting trajectories was the overall time to reach the target altitude, which corresponded to the change in thrust for the different engines. All three trajectories agree strongly with the published world record performance of the F-4 from 1962, which demonstrates the validity of the models.

Overall, these results demonstrate that even for a relatively simple problem coupling is important between the engine and trajectory analyses. Furthermore, the implementation demonstrated that the method supports parallel execution of the propulsion system model and trajectory analyses for each node and that it is computationally feasible to run the expensive propulsion analysis model within the trajectory optimization itself. Although this work only used an improved propulsion model and left the aerodynamics model as a low fidelity approximation, these results show that the work could be extended to include higher-order aerodynamics models (e.g. panel methods or CFD) as well.

While the results presented in the paper show promise for the tightly coupled approach, additional work remains to fully evaluate the methodology. Primarily, the method needs to be applied to a more complex problem that would include more engine controls that don't have a trivial and obvious solution such as inlet bleed schedules, variable area nozzles, and variable geometry turbomachinery. Beyond including more control

parameters with non-trivial solutions, the problem should also include engine variables such as compressor pressure ratio and engine sizing. Including a more broad set of control and design variables would allow for the engine to be sized to produce the required thrust based on both the trajectory and control schedule. Finally, a more detailed aircraft sizing could eventually be included in evaluation of the methodology, with the optimizer potentially determining some of the aircraft design variables such as wing area and aspect ratio.

Acknowledgments

The work presented in this paper was supported by NASA's Transformational Tools and Technology Project. The authors would also like gratefully acknowledge Dr. Jonathan Burt at NASA Glenn Research Center for helping to implement the LGR transcription in Pointer and Jeff Chin, Sydney Schnulo, Tristan Hearn, and Ken Moore for their contributions to the development of PyCycle.

References

- ¹Anderson, J., *Aircraft Performance and Design*, WCB/McGraw-Hill, 1999.
- ²Roskam, J., *Airplane Design: Preliminary Sizing of Airplanes*, Airplane Design, DARcorporation, 2003.
- ³Welstead, J. R. and Felder, J. L., "Conceptual Design of a Single-Aisle Turboelectric Commercial Transport with Fuselage Boundary Layer Ingestion," *54th AIAA Aerospace Sciences Meeting, San Diego, CA*, 2016.
- ⁴Drela, M., "Development of the D8 Transport Configuration," *29th AIAA Applied Aerodynamics Conference, Honolulu, HI*, 2011.
- ⁵Bradley, M. K. and Droney, C. K., "Subsonic Ultra Green Aircraft Research: Phase I Final Report," Contractor Report NASA/CR-2011-216847, NASA, April 2011.
- ⁶Bradley, M. K. and Droney, C. K., "Subsonic Ultra Green Aircraft Research: Phase II Volume II Hybrid Electric Design Exploration," Contractor Report NASA/CR-2015-218704, NASA, April 2015.
- ⁷Felder, J. L., Kim, H. D., and Brown, G. V., "Turboelectric distributed propulsion engine cycle analysis for hybrid-wing-body aircraft," *47th AIAA Aerospace Sciences Meeting, Orlando, FL, January*, 2009.
- ⁸Tavares, T., "A Supersonic Fan Equipped Variable Cycle Engine for a Mach 2.7 Supersonic Transport," Contractor Report NASA/CR-177141, NASA, August 1985.
- ⁹Gray, J., Mader, C. A., Kenway, G. K. W., , and Martins, J. R. R. A., "Approach to Modeling Boundary Layer Ingestion using a Fully Coupled Propulsion-RANS Model," *55th AIAA Aerospace Sciences Meeting (SciTech)*, Grapevine, TX, January 2017.
- ¹⁰Hearn, T., Hendricks, E., Chin, J., Gray, J., and Moore, K., "Optimization of Turbine Engine Cycle Analysis with Analytic Derivatives," *AIAA Aviation Forum 2016*, June 2016.
- ¹¹Falck, R., Gray, J., and Naylor, B., "Parallel Aircraft Trajectory Optimization with Analytic Derivatives," *AIAA Aviation Forum 2016*, June 2016.
- ¹²Gray, J. S., Moore, K. T., and Naylor, B. A., "OpenMDAO: An Open-Source Framework for Multidisciplinary Analysis and Optimization," *13th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Fort Worth, TX, AIAA, AIAA-2010-9101*, AIAA, Fort Worth, Texas, August 2010.
- ¹³Martins, J. R. R. A. and Hwang, J. T., "Review and Unification of Methods for Computing Derivatives of Multidisciplinary Computational Models," *AIAA Journal*, Vol. 51, No. 11, 2013, pp. 2582–2599.
- ¹⁴Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Journal on Optimization*, Vol. 12, 2002, pp. 976–1006.
- ¹⁵Gray, J., Chin, J., Hearn, T., Hendricks, E., Lavelle, T., and Martins, J. R. R. A., "Thermodynamics of Gas Turbine Cycles With Analytic Derivatives in OpenMDAO," *54th AIAA Aerospace Sciences Meeting*, 2016.
- ¹⁶"U.S. Standard Atmosphere, 1976," Tech. Rep. NASA-TM-X-74335, National Oceanic and Atmospheric Administration, National Aeronautics and Space Administration, United States Air Force, October 1976.
- ¹⁷Garg, D., Patterson, M. A., Francolin, C., Darby, C. L., Huntington, G. T., Hager, W. W., and Rao, A. V., "Direct trajectory optimization and costate estimation of finite-horizon and infinite-horizon optimal control problems using a Radau pseudospectral method," *Computational Optimization and Applications*, Vol. 49, No. 2, 2011, pp. 335–358.
- ¹⁸Fahroo, F. and Ross, I. M., "Pseudospectral methods for infinite-horizon nonlinear optimal control problems," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 4, 2008, pp. 927–936.
- ¹⁹Herman, A. L. and Conway, B. A., "Direct optimization using collocation based on high-order Gauss-Lobatto quadrature rules," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 3, 1996, pp. 592–599.
- ²⁰Geiselhart, K. A., Ozoroski, L. P., Fenbert, J. W., Shields, E. W., and Li, W., "Integration of Multifidelity Multidisciplinary Computer Codes for Design and Analysis of Supersonic Aircraft," *49th AIAA Aerospace Sciences Meeting*, No. AIAA-2011-465, 2011.
- ²¹Bryson, A. and Denham, W., "A Steepest-Ascent Method for Solving Optimum Programming Problems," *Journal of Applied Mechanics*, ASME, 1962.
- ²²Hargraves, C. and Paris, S., "Direct Trajectory Optimization Using Nonlinear Programming and Collocation," *Astrodynamics Convergence*, Williamsburg, VA, 1982.
- ²³Bryson, A. E., *Dynamic Optimization*, Addison Wesley Longman, 1999.

²⁴Fédération Aéronautique Internationale, “Powered Aeroplanes World Records,” <http://www.fai.org/record-powered-aeroplanes>, ID:9068.

²⁵Daly, M., *IHS Jane’s Aero-Engines*, Vol. 27 of *Jane’s Aero-Engines Series*, Jane’s Information Group, 2010.

²⁶Kulas, M. J., “Propulsion Characteristics Summary,” Tech. Rep. NAVAIR 00-110A-3, United States Navy, August 1992.